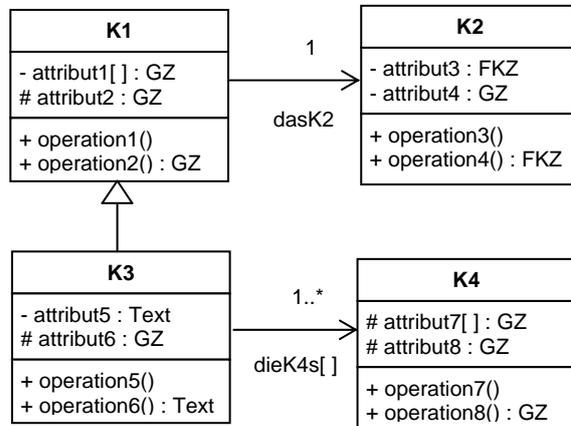


1 Grundlagen der Objektorientierten Programmentwicklung

Punkte

1.1 Nennen Sie die Beziehungen, Eigenschaften und Verhaltensweisen eines Objektes der Klasse K3.

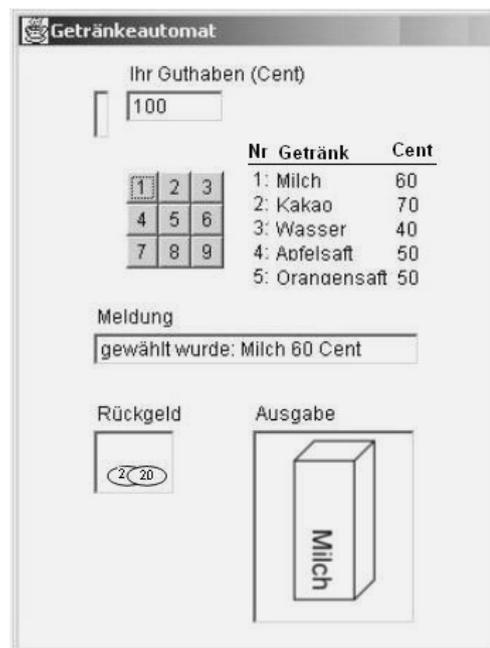


3

2 Getränkeautomat

Ein einfacher Getränkeautomat soll programmiert werden.

Der Automat akzeptiert Münzen im Wert von 10,20,50 Eurocent, 1 und 2 Euromünzen und wird wie folgt bedient:
Der Bediener wirft eine oder mehrere Münzen in den Münzeinwurf bis das Guthaben mindestens den Preis des Getränkes erreicht hat. Das Guthaben wird neben dem Münzeinwurf angezeigt. Die eingeworfenen Münzen werden automatisch in die Kasse des Automaten übertragen. Danach kann man am Ziffernfeld das gewünschte Getränk über die Eingabe der Getränkenummer wählen (siehe Abbildung). Ist das Getränk im Lager vorhanden erscheint im Textfeld Meldung das gewählte Getränk und der Preis des Getränkes. Das Getränk wird in diesem Fall vom Lager in den Ausgabeschacht befördert; das Rückgeld wird berechnet und von der Kasse in den Rückgeldschacht gegeben. Ist das gewünschte Getränk nicht im Lager vorrätig, erscheint eine Information im Textfeld Meldung, und das Guthaben des Kunden wird von der Kasse in den Rückgeldschacht übertragen.



Folgende Klassen sind für die Modellierung der Software ausreichend:

Ein **Steuerungsobjekt** speichert im Attribut *dieWarenNr* die zu dem gewählten Getränk gehörige Nummer. Die Operation *uebergebeAnKasse()* informiert die Kasse, welche Münze eingeworfen wurde. Über die Operation *gewaehltWurdeWare()* wird ein Steuerungsobjekt informiert, welches Getränk gewählt wurde. Die Operation *berechneRueckgeld()* ermittelt die Differenz zwischen dem Guthaben und dem Getränkepreis und veranlasst das Ermitteln und die Herausgabe des Rückgeldes.

Steuerung
- dieWarenNr : GZ - derPreis : GZ
...
+ uebergebeAnKasse(pMuenzNr : GZ) + gewaehltWurdeWare(pWareNr : GZ)

Punkte

Eine **Kasse** speichert im Attribut *dasGuthaben* den Gesamtwert der eingeworfenen Münzen. Die Attribute *dieMuenzWerte[]* und *derVorratAnMuenzen[]* dienen dazu, den Münzvorrat zu verwalten.

Kasse
- dieMuenzWerte [] : GZ - derVorratAnMuenzen [] : GZ - dasGuthaben: GZ
+ erhoeheGuthaben(pMuenzNr: GZ) + gibGuthaben() : GZ + gibRueckgeld(pRueckgeld : GZ)

Feld *dieMuenzWerte[]*:

Münzwert(Cent)	10	20	50	100	200
Feldindex = Münznummer	0	1	2	3	4

Die Operation *gibRueckgeld()* ermittelt die Münzen, die der Bediener zurückbekommt und gibt diese Münzen in den Rückgeldschacht.

Im Attribut *derVorratAnWaren* verwaltet ein **Lagerobjekt** die Menge an vorrätigen Getränken. Der Parameter *pWareNr* der Operationen entspricht den Getränkeummern, die den Getränken am Automat zugewiesen wurden. Ein Lagerobjekt bewirkt das Ausgeben des Getränkes in den Ausgabeschacht.

Lager
- derVorratAnWaren [] : GZ
+ gibWareAlsObjekt(pWareNr : GZ) : Ware + istWareVorhanden(pWareNr) : Boolean + gibBezeichnung(pWareNr: GZ) : Text + gibPreis(pWareNr: GZ) : GZ + gibWareAus(pWareNr: GZ)

Weiterhin steht die Klasse **Ware** zur Verfügung.

Ware
- dieBezeichnung : Text - derPreis : GZ
+ Ware (pBezeichnung: Text, pPreis: GZ) <<constructor>> + gibBezeichnung() : Text + gibPreis() : GZ

Zur **Oberfläche** gehören die Tastatur, die Guthabeanzeige, das Textfeld Meldung und der Schlitz für den Münzeinwurf. Die Mitteilungen im Textfeld Meldung werden über die Operation *schreibeMeldung()* ausgegeben.

Oberflaeche
...
+ Oberflaeche(pSteuerung : Steuerung) <<constructor>> + schreibeMeldung (pMeldung : Text) ...

Anmerkung: Standardkonstruktoren sind nicht angegeben.

- 2.1 Die Klasse Steuerung besitzt Operationen mit unterschiedlichen Sichtbarkeiten. Erklären Sie, wie sich dieser Unterschied auf die Verwendung der Operationen auswirkt. 1
- 2.2 Erstellen Sie ein vollständiges Klassendiagramm ohne Angabe der Attribute und Operationen. Verwenden Sie nur die Klassen, die oben dargestellt sind. Geben Sie bei den Assoziationen die Kardinalitäten und Rollennamen an. Begründen Sie jeweils kurz, weshalb Sie sich für die eingetragene Kardinalität und Richtung der Assoziation entschieden haben. 6

	Punkte														
2.3 Die Klasse Lager besitzt die Operation <i>gibWareAlsObjekt(pWareNr : GZ) : Ware</i> . Erklären Sie, wozu diese Operation dient.	1														
2.4 Modellieren Sie das folgende Szenario in Form eines Sequenzdiagramms: Der Kunde wirft eine 1-Euromünze ein und wählt anschließend Milch über das Drücken der Taste 1. Gehen Sie davon aus, dass noch ausreichend Milch im Lager vorhanden ist. Die Meldung "gewählt wurde: Milch 60 Cent" wird ausgegeben. Der Kunde erhält die Milch und das Rückgeld. Tragen Sie bei den Botschaften die <u>konkreten</u> Parameterwerte und Rückgabewerte ein. Beginnen Sie mit der Nachricht <i>gewähltWurdeWare(1)</i> an das Steuerungsobjekt.	6														
3 Zustandsdiagramm für ein Tastaturobjekt Die Tastatur eines beliebigen Automaten ist so lange im Zustand 'bereit', bis ein Bediener eine Taste drückt, um entweder eine Ware zu wählen oder den Zustand 'öffne Tür' herbeizuführen, um z.B. das Lager aufzufüllen. Folgt nach dem erstmaligen Drücken einer Taste innerhalb von 2 Sekunden kein weiterer Tastendruck, kommt das Objekt in den Zustand 'Ware gewählt'. Nach Ausgabe der Ware und des Rückgeldes befindet sich die Tastatur wieder im Zustand 'bereit'. Berücksichtigen Sie <u>nicht</u> , dass eventuell Ware oder Rückgeld fehlen. Will der Bediener den Zustand 'öffne Tür' erreichen, ist es erforderlich, die Nummernfolge 999 so einzutippen, dass zwischen dem Drücken der Tasten maximal 2 Sekunden Zeit verstreichen. Nach dem Schließen der Tür wird wieder der 'bereit'-Zustand eingenommen. Erstellen Sie ein Zustandsdiagramm für das Tastaturobjekt. Transitionen können durch freie Texte beschrieben werden.	6														
4 Algorithmus zur Ausgabe des Geldbestandes Die Klasse <i>Kasse2</i> besitzt die Operation <i>berechneGeldbestand(pMagazin : GZ) : GZ</i> . Die Operation ermittelt den Geldwert der Münzen in den Münzmagazinen. Der Parameter <i>pMagazin</i> gibt dabei das Magazin an, dessen Geldwert ermittelt werden soll. Besitzt <i>pMagazin</i> den Wert 99 wird der Gesamtgeldwert aus allen Magazinen ermittelt.	7														
<table border="1"> <tr> <td>Geldwert des Münzmagazins (Cent)</td> <td>10</td> <td>20</td> <td>50</td> <td>100</td> <td>200</td> <td>10,20,50,100,200</td> </tr> <tr> <td><i>pMagazin</i></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>99</td> </tr> </table>	Geldwert des Münzmagazins (Cent)	10	20	50	100	200	10,20,50,100,200	<i>pMagazin</i>	0	1	2	3	4	99	
Geldwert des Münzmagazins (Cent)	10	20	50	100	200	10,20,50,100,200									
<i>pMagazin</i>	0	1	2	3	4	99									
Das Feld <i>dieMuenzWerte[]</i> verwaltet die Münzwerte der Magazine:															
<table border="1"> <tr> <td>Münzwert (Cent)</td> <td>10</td> <td>20</td> <td>50</td> <td>100</td> <td>200</td> </tr> <tr> <td>Index</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> </table>	Münzwert (Cent)	10	20	50	100	200	Index	0	1	2	3	4			
Münzwert (Cent)	10	20	50	100	200										
Index	0	1	2	3	4										
Das Feld <i>derVorratAnMuenzen[]</i> verwaltet die Anzahl der in den Magazinen vorrätigen Münzen. Wenn sich z.B. sieben 1-Euro-Münzen in der Kasse befinden, hat die Variable <i>derVorratAnMuenzen[3]</i> den Wert 7.															
Entwickeln Sie den Algorithmus der Operation <i>berechneGeldbestand(pMagazin : GZ) : GZ</i> und stellen Sie diesen in Form eines Struktogramms dar.															

